A SELF ESTABLISHING CLUSTERED NETWORK ARCHITECTURE FOR BLOCKCHAIN

A Thesis

by

Orkun Doğan

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Computer Science

Özyeğin University
December 2022

A SELF ESTABLISHING CLUSTERED NETWORK ARCHITECTURE FOR BLOCKCHAIN

Approved by:

Assistant Professor Kübra Kalkan, Advisor
Dept. of Computer Science
Özyeğin University

Associate Professor Hasan Sözer
Dept. of Computer Science
Özyeğin University

Professor Fatih Alagöz
Dept. of Computer Eng.
Boğaziçi University

Date Approved: 20 December 2022

Dedicated to my family who have pushed me towards success and greater achievements my whole life...

ABSTRACT

Blockchain technology has branched out into many industries, such as healthcare, manufacturing, agriculture and entertainment, in the shape of both of its public and non-public variants. In principle, blockchain provides these industries with an immutable ledger, allowing the processes in its application environment to be taken care of in a decentralized manner. However, some challenges blockchain has faced to this day remain, such as the degree of its scalability, the level of security it provides and the transparency of the network transactions. In this thesis, a novel approach to a distributed, permission-less blockchain network is explored with the use of hierarchical clustering to gather the nodes based on the latency of their connection to one another. These clusters of nodes are allowed to work on their respective local chains and to add the verified local chains to the actual global chain that is used by the entire system. Network's throughput performance and overall latency are evaluated and compared with other blockchain applications, namely a simulation of the Bitcoin network itself and another approach that makes use of a method called Community Clustering. We collected the data for the correlation in the same environment for our work, Bitcoin and Community Clustering[1] networks. The comparison of the collected data aligns with our work's clusters to improve the transaction throughput of the network, where an increase in average throughput and a drastic decrease in latency are observed.

ÖZETÇE

Blok zinciri teknolojisi gelişerek ve genişleyerek sağlık, imalat, tarım ve eğlence gibi birçok endüstriye, herkese açık ve özel varyantları olacak şekilde ulaşabilmiştir. Temelinde blok zinciri teknolojisi bu endüstrilere uygulama ortamında bulunan işlemlerin merkezsizleştirilmiş bir biçimde oluşmalarını sağlayan değişmez bir kayıt defteri sunar. Ancak blok zinciri teknolojisinin geçmişten günümüze karşılaştığı bazı zorluklar halen varlığını sürdürmektedir. Bu zorluklardan bazıları blok zincirinin ölçeklenebilirlik derecesi, sunuyor olduğu güvenlik seviyesi ve ağda meydana gelen işlem ve hareketlerin saydamlığı olarak sayılabilir. Bu tezde dağıtık, izin gereksinimi olmayan blok zinciri temeline sahip bir ağ yapısına düğümlerin (ağ katılımcılarının) aralarındaki bağlantı gecikmesini esas alan hiyerarşik kümelenme kullanılarak en düşük olanlar birlikte tutulacak şekilde yeni bir yaklaşım araştırılmıştır. Bu düğüm kümelerinin kendi yerel zincirleri üzerinde çalışmaları sağlanıp, ağın kalanı tarafından da onaylanan yerel zincirlerin sistemin geneli tarafından kullanılan global zincire eklenmeleri sağlanmıştır. Ağın verimliliği, ağda işlenebilen iş hacmi ve ağın genel gecikme süreleri değerlendirilmiş ve diğer blok zinciri uygulamaları, Bitcoin[2] ve Komünite Kümelemesi (Community Clustering)[1] isimli bir metod kullanan başka bir yaklaşım ile kıyaslanmıştır. Öne sürdüğümüz yöntem, Bitcoin ve Komünite Kümelemesi ağlarının kıyaslanması için gerekli verileri aynı ortamda toplanmıştır. Toplanan verinin kıyaslanmasından elde edilen sonuçlar öne sürdüğümüz yöntemin işlem hacmini iyileştirmeyi hedefleyen kümeleri ile hizalanmakta olup, ağdan elde edilen ortalama işlem hacminde bir artış ve gecikmede de şiddetli bir düşüş olduğu gözlemlenmiştir.

ACKNOWLEDGEMENTS

I would like to acknowledge the advisor of my thesis, Assistant Professor Kübra Kalkan, for all the advice, guidance and support she has provided me during my graduate studies, the development and realization of my master's project and the writing of this thesis and show my gratitude to her. Where I waned, she gave me courage and put me back on the right path by encouraging me to work harder despite the circumstances. Her vast knowledge and experience in the field played a crucial part in my studies and allowed me to draw a path so that I could make this thesis a reality.

I also would like to show my gratitude to all the academics in the committee, Associate Professor Hasan Sözer and Prof. Dr. Fatih Alagöz, for their interest, time and comments on my thesis.

Lastly, I would very much like to show my utmost gratitude to my parents and my brother. Without their unwavering support throughout my life, this thesis would never have been possible. Through thick and thin I have always felt their support and its mere presence still gives me courage to push forward in life. I hope to be able to make the most of the support they have shown me all my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The blockchain technology has obtained a foothold in the cryptocurrency market as it ensures security, transparency and trust for the participants of a network. However, blockchain is not defined only to currencies and provides more than just a means of dealing with money. The root promise is the elimination of the central trustee in the structure. In the internet, the trust is provided by banks for financial transactions, by the government and courts for legal actions. By eliminating the trust provider, the blockchain disperses confidence among the system by mining, formed in a distributed manner by the collective efforts. This allows the users the ability to carry out actions or access to means of purchasing goods and services in a decentralized manner, a much wider range of industries than just cryptocurrencies to make use of the technology such as entertainment or even healthcare[3, 4, 5, 6, 7, 8].

1.1 Blockchain background

1.1.1 Types of blockchain

The number of main blockchain types that are used is four, constituting of public, private, consortium and hybdrid blockchains. Private blockchains are used in restricted network environments, the participants need permission from the governing entity to join and operate in these types of blockchain networks. Hybrid blockchain networks consist of two different environments one of which is a public blockchain and the other is private. In such environments the freedom of the participant entities is still restricted by a governing entity, their access to certain data and even access to the private chain by itself is up to the governing entity of the network. Consortium blockchain networks also have both public and private chains included in the network,

however, there are also participant nodes that are determined beforehand that are tasked with the management of the consensus algorithm of the whole network. This also means that the trust and security of the network is handed over to these management nodes. The scope of this thesis is focused on public blockchain networks. This kind of blockchain network promises their participants the highest degree of freedom while keeping the integrity, trust, security and privacy of the network intact. The participants may join, leave and contribute to the efforts of the network as they like due to the self-governed and decentralized nature of the network. Self-governance and decentralization is achieved through the collective consensus of the network instead of a singular entity that usually provides, audits and manages the means of trust between the participant peers. In contrast to these public ones, there are private blockchain networks, which work in a similar way in terms of the kind of data distribution scheme, which is a distributed system, however, they differ in the sense that in order for an entity to join and participate in these private networks, they first require authentication and authorization from singular and central beings that manage and govern the network.

1.1.2 Consensus algorithms

Consensus algorithms are processes used by distributed systems in order to achieve a collective agreement on the integrity and trustworthiness of any given data value. In blockchain systems these algorithms are used in order to achieve a consensus among the participants that their participation is genuine and trustworthy. The consensus algorithms are employed in order to verify the integrity and reliability of the contributions of the peers, therefore establishing the trust needed between the peers which allows the network to function. There are mainly two consensus algorithms used in blockchain networks. One is named Proof of Stake (PoS), where the energy usage, processing time and computational cost is minimal compared to other algorithms. In

this consensus scheme, the participants use their network tokens to raise their chances of becoming the next validator for a newly generated block where the participant who has staked the largest amount of tokens gets the highest chance of becoming the next validator. The other consensus algorithm is called Proof of Work (PoW), where the participants use their processing and computational power to solve complex mathematical problems, which also means the energy cost of these networks is quite high. The scope of this thesis is focused on the Proof of Work algorithm. In Proof of Work algorithms, the participant peers use their hardware resources in an algorithm that aims to find a random hash value which is calculated with the contents of a created block that passes the difficulty check of the network. Whenever a participant creates a block with an acceptable hash value, it sends that block to the rest of the network so that everyone else can verify the block. The verified blocks are then accepted into the globally used chain as the last block to be added to it. This process allows the network to function securely.

1.1.3 Bitcoin

Bitcoin [2], the flagship application of the blockchain technology itself is of course considered while exploring the approach presented in this thesis. As explained in [9], the Bitcoin network uses a Proof of Work consensus algorithm where the network participants are required to calculate hash values until they find one that is equal to or lower than the value that is required by the structure. Nodes trying to discover the expected hash values are called *miners* and the calculations they do are called *mining*. The point of this mining procedure is for the participants to prove that they are fair nodes doing decent work and contributing to the system. This Proof of Work algorithm is used to counteract the impact the dishonest or malicious nodes, who wish to manipulate the network to their advantage, might have on the network's consensus.

1.1.4 Proof of Work shortcomings

Blockchain networks utilizing the Proof of Work consensus algorithms suffer from two deficiencies, scalability and the transaction throughput of the network. The throughput are insignificant because using Proof of Work algorithms take time and computational effort to reach a consensus. There is also a need for higher throughput values specifically for businesses that expect fast payment. These kinds of networks also require high scalability to allow the usage of the network by a wider range of users. Our work addresses both the scalability and throughput issues at the same time. In the Bitcoin network, the difficulty of the network is adjusted so that a block is mined every 10 minutes. This was chosen as a way of adjusting the tradeoff between first confirmation of a block and the subsequent branches that may appear because of these blocks. Mining too many blocks in a short amount of time causes many branches to appear and having too many branches might mean wasting a huge amount of processing power if nodes were to keep on mining on the branches that were not to be accepted in the future. This branching issue is addressed in a different manner with our work. Due to the dynamics of the network, there may be blocks that have been mined but were left unaccepted at the end of their verification process, resulting with orphan blocks appearing in the network. Depending on the specifics of the implementation of the work presented in this thesis this may or may not cause forking in the blockchain itself. In this thesis, these orphan blocks or even orphan local chains are simply discarded and are simply disregarded for further mining. This branching issue happens in the Bitcoin network [2] as well and this network also discards the branches that may appear from time to time, however, the discarding of the non-canonical chains only happens if one chain simply becomes longer if the nodes mine on it and can then propagate their blocks through the network. The nodes have to race one another in order to get the rest of the network to accept the branch they were mining on as the canonical branch. It also means that all the processing power

that was used on the non-canonical branches and all the time that has gone into that branch have been wasted, even though all these resources could have been handled in a better way if the branching issue was handled earlier. Which is what our work aims to do, by eliminating the branches right when they appear in order to keep the processing power and the rest of the network resources focused on achieving a unified effort on the canonical chain, instead of making the nodes waste resources on chains that may end up discarded down the line.

1.2 Self-Establishing Blockchain Network

The work presented in this thesis is an architecture use of blockchain structure and aims at exploring an advanced method on the usage of particular network parameters, distributed hierarchic clustering and introducing smaller chains that are local to every cluster in the system, accordingly constituting a blockchain system appropriately adjusted for the structures require transaction rates much higher than the prominent cryptocurrency systems support. Healthcare systems and entertainment industries are two prime examples of such networks. Transaction amount increment is provided by use a hierarchical clustering algorithm that provides the nodes the means to identify and introduce themselves into clusters that are best suited for both themselves and the system. In doing so, this approach reduces the amount of network connections between the associates as well as the network congestion[10][11][12]. The fundamental metric used to figure out the network performance is transactions per second (Tps), standing for the number of transactions per second the network accepts in global blockchain. Network also emphasizes configurable parameters that impact the network performance[13]. These parameters, according to the type and importance of their impact, can be considered key points in adjusting the performance of the network[14].

In our work, a node goes through three phases. Initial phase is where nodes seek

other peers, second step is the gathering where the nodes cluster both themselves and each fits to best clusters, third step is the mining step where all the nodes in all the clusters prove their practice. At this stage, the mined blocks are added to the *local chains* that are accepted by their clusters, and when these chains reach a certain length, they broadcast to the network so they can be verified. Every cluster in the network has its own local chain, which is worked on by all the members of the local chains' respective cluster. Aside from these local chains there is also a singular global chain that is shared among all the members contributing to the efforts of the network. When the local chains are verified by the system, they are then added to the end of the global blockchain.

The contributions of this thesis can be listed as follows:

- Introducing a distributed and decentralized version of Hierarchical Clustering among the participant nodes of the structure
- Dividing the workload among the members of clusters with locally mined smaller blockchains of adjustable sizes and allowing a considerably efficient use of both the collective computing capacity and the networking resources
- Designing a decentralized permissionless blockchain network utilizing a Proof of Work algorithm while having a high transaction rate that can be used in certain industries such as healthcare or entertainment
- Collecting data from both the developed program and the simulated Bitcoin network, analyzing the data in order to evaluate how the transaction rate metric of the network performs and compares against the simulated Bitcoin network

A network is considered to be permissionless when nodes searching to join the network are not expected to seek permission from a central being that would grant them a kind of authentication allowing the nodes to join the network and communicate

with other members of the said network. Meaning, in our work the nodes can join and leave the network at their own discretion.

The rest of the thesis is structured as follows. Chapter II provides a detailed literature review. Chapter III explains the three phases and the network architecture. Section IV presents the performance of our work according to the data gathered from its experimental runs together with the comparison of our work to the simulated Bitcoin and community clustering networks in the same environment. The thesis is concluded in section V.

CHAPTER II

LITERATURE REVIEW

In this section we provide a detailed literature review on clustered blockchain network applications.

2.1 Previous Work

Fadhil et. al. focus on the use of clusters in the blockchain networks [15]. In their approach the network architecture consists of units they call Super Peers. Also, not all participants of their network contribute to the consensus of the network. Which in turn means that there is a considerable amount of network resources being wasted even though they could have been used more efficiently and effectively to improve the performance of the network. Their approach to the network architecture poses a centralization threat where any non-conforming Super Peer gains the upper hand in the network communications, therefore, posing a threat to the security and integrity of the whole network if not reducing its performance at the very least by feeding false or forged information to the other participants. On the other hand, in our work, a more decentralized approach is utilized.

Alternatively, Li et. al. [1] have taken an approach similar to the one used in our work, where they have made it so that the nodes would be clustered in what they call *communities*, however, in their blockchain structure, each community has its own chain but there is no global blockchain that is normally used by all the nodes in the network. In their paper it is stated that their approach seems to have reduced the overall network load as well as the inter-cluster communications needed between the peers, therefore improving the efficiency of the network. Such results from other approaches using clustering is quite heartening for our work's success as

well. However, their approach falls short where a real world application needs a global blockchain to be used by all the clusters in the network, such as the healthcare system.

Jin et. al. [16] have also made use of clusters in the network, however, they also included a centralized learning system called Federated Learning. Since Federated Learning has flaws when it comes to decentralized networking, such as central server malfunctions, they propose a new approach to Federated Learning called Cross-cluster Federated Learning (CFL). Their claim is that this new approach addresses the data sparsity and high latency communications in the blockchain-based Federated Learning (BFL). This new federated learning approach seems similar to the approach used in our work, in the way that instead of using one big cluster of nodes, CFL also uses smaller and numerous clusters of nodes in the network. The difference is that their approach is focused more on using and training the Machine Learning algorithm, instead of actually making use of the Proof of Work algorithm all the while having a high transaction rate, whereas our work is more focused on the actual processing of the transactions in the network.

Another approach that makes use of clustering in the blockchain networks was presented by Biryukov et. al. [17], where they have clustered the transactions instead of the nodes themselves. In their approach, the transactions also carry a timestamp from the originating node and the IP addresses of the nodes are assigned weights. These weights are then used to calculate weight vectors in order to understand the relations between the propagated transactions. The weight vectors are calculated with a parameterized weight equation that takes into account the lessening importance of the number of times a transaction is propagated. The clustering they have implemented is highly focused on making use of the transactions rather than the clustering of the participant nodes, this difference is important because clustering the transactions would not prove to be useful in terms of the transaction rate of the network. It also means that their network structure remains unaffected by the clustering algorithm,

therefore the actual network communications between the nodes are also unaffected which in turn means the load on the network and the network congestion still remain the same. However, our work's aim is to reduce the network congestion in order to improve the transaction rate, so the work presented in [17] does not include any of the priorities of our work.

Saadat et. al. [18], had the aim to create a network architecture in a way that allows the network to either be merged or split according to the needs of the network. Their approach to the use of clustering in the blockchain networks is more focused on the usage of the clusters themselves rather than using the clusters of nodes as a means of reducing the network load to improve performance. In other words, the aim was to allow the reconfiguration of the network to redistribute the participants in new or different clusters, instead of improving the overall network performance and increasing the transaction rate of the network.

Islam et. al. [19] propose a secure and distributed network architecture that also makes use of clustering. However their architecture uses units they call *cluster heads*, they have even introduced a specialzed *Cluster Head Selection* algorithm just for this purpose. The reason for this is that their network is mainly focused on reducing the energy consumption, not the transaction rate of the network. Using a cluster head also introduces possible attack vectors for malicious parties however, if a selected cluster head somehow decides to act malicious then the whole network would be affected by its actions. However, since their target environment is IoT infrastructures, having cluster heads might prove useful as well. Since this is not the case with our work, instead of using cluster heads, the responsibility of organizing the network has been distributed to all the nodes in the network in our work.

Joshi et. al. [20] focus on a clustering based data transmission solution for vehicular networks, called VANET. The authors of this work propose an architecture called ROAC-B which also makes use of clustering for the participant nodes in the

network, in this case the participant nodes are the intelligent vehicles in the network. They use clustering for the purposes of load balancing and minimizing network overhead. The same can be said for our work, since it makes use of clusters as well, therefore indirectly balancing the network flooding, limiting the effect flooding has on the network.

Pajooh et. al. [21] make use of clustering in order to provide a multi-layered architecture to their network. They use a so-called IoT network clustering, which is a clustering algorithm they have adapted in order to make the clustering more suitable for the needs of the IoT network they work on. Their adapted clustering provides security as well as reduced network load and improved the network coverage at the same time. The fact that their clustering provides a reduced network load is in line with what our work aims for, however, in their approach the clusters have what's called "cluster heads". Cluster heads are made to be responsible for the local authentication and authorization of their clusters. In our work however, clusters do not have any central management nodes, instead every node is treated the same and they have both the same responsibilities and the same privileges.

Mittendorfer et. al. [22] propose a clustered blockchain architecture for IoT enviroments in their work. Their clusters consist of the end points of the IoT infrastructure, typically the sensors. These clustered sensors are only tasked with sending their respective data to the cluster manager, this manager is where their network becomes centralized as the role of the manager is the block creation. Instead of the sensors, these cluster managers are the sole blockchain participants from their cluster and they represent their respective clusters on the network. Their proposed work does reduce the network flooding effect of continuous messaging between the nodes, however, their network architecture is not fully centralized. In our work, clustering is also utilized to reduce network flooding, however, this is done in such a way that the network remains decentralized with no centralized participants in the blockchain

network.

In [23] Pajooh et. al. propose a network architecture for blockchain networks with a multi-layered security mechanism. In their architecture there are Cluster Heads that are treated as powerful devices that can handle the distributed communication of the blockchain network. These Cluster Heads prove to be centralized units inside the network whereas in our work, all nodes are treated equally as participants of the network. Our work, free from centralized network participants, uses a clustering method as they do in their proposed architecture in order to provide better usage of network resources.

In [24], Amiri et. al. propose a clustered blockchain network where the aim is to properly distribute the nodes into clusters. They do so by placing the nodes into clusters according to their geological locations in the sense that the nodes that are physically closer to one another are clustered together. In our work however, the placement of nodes into clusters is handled differently and in a way that also requires the consensus of the network. The nodes use their connection quality as a metric to place themselves into a cluster by averaging their connection quality to the rest of the cluster they wish to be a part of. Physical proximity between nodes do not necessarily mean a better connection quality, therefore our work aims to outperform their proposed approach by using more cleverly designed connection quality metrics. This process indicates that some nodes may send faulty data to the rest of the network therefore it is also verified by the network and thus the nodes are clustered with the consensus of the network rather than any centralized participant or the sole will of any given participant node.

Fadhil et. al. [25] propose a locality based approach to clustering in order to improve the propagation of information on the Bitcoin network. They state that reducing the connection cost geographically also reduces the propagation delay between the nodes. In our proposed network architecture connectivity measurements

are used instead of geographical measurements, meaning that nodes that might be further away from each other with better connection measurements will be placed into the same cluster rather than the physically closest nodes. This ensures that the actual communication is taken into account when working towards improving the transaction throughput and the propagation of information inside the network.

The related works mentioned are similar to our work in the sense that they also venture into the effects of clustering on blockchain networks, however, they remain different from our work. Some of the differences are such that in some of the works the effort is focused on centralized blockchain networks whereas our work is fully decentralized and in others some contributors of their respective networks are neglected and are unable to participate, therefore resulting in a loss of total network capacity usage, whereas our work aims to allow all contributors to contribute to the consensus of the network. On top of which our work utilizes the network resources to the maximum, all the while making sure that all transactions are treated equally and processed properly without relying on a centralized network management unit.

CHAPTER III

SECBON ARCHITECTURE

The name of our work is SECBoN which stands for Self-Establishing Clustered Blockchain Network. SECBoN is a decentralized system structure having no central participant or coordinator nodes. All the participants work collectively, just as required from a distributed structure require for the consensus. However, SECBoN also introduces a collection of arrangements to both its network and chain structure. Participants are gathered corresponding to the connection quality, the nodes have low latency link to one another, are bundled closely. The metric to evaluate the quality of the communication here is based on the latency between the nodes. At first, clusters have their own local chains and the members of these clusters mine blocks on their respective local chains. When particular conditions are met, these local chains are propagated through the network for verification. The local chains that have been verified are then added to the global blockchain. This approach provides a boost in mining speed as the mined blocks will be propagated inside a cluster first, meaning the network congestion will be handled in a better way, as opposed to how Bitcoin handles it and the broadcasted blocks won't cause as much of a network congestion as they do in other networks. Reduced number of connections between different clusters means that fewer connections with poor quality are used. Using a lesser number of connections with poor quality and a higher number of connections with better quality means the network's resources will be used in a smarter and more efficient manner. Whether or not the local chains are valid or legitimate is still decided by the whole network since these short chains are verified by all participants, meaning the trustworthiness of the global chain is kept intact and any cluster could not act in

a malicious manner on its own.

3.1 Network Topology

One of SECBoN's defining features is the self-forming network structure. During the first step of the network setting up, the nodes try to find other nodes within a range of IP addresses. This process is carried out by trying unique addresses to find a node before the time runs out. The network structure itself, however, forms on its own as the nodes discover one another on the network. The resulting network structure is that of a Mesh Network. However, in a proper Mesh Network, all the nodes would be connected to one another, which is not the case in SECBoN. This is because nodes try to find others in the network in a limited time frame, therefore they may not find all other participants, resulting in a connected Mesh Network. The time frame for this step is also a parameter of the network, meaning this step has performance related implications which will be covered in the upcoming parameters section, Section IV, of this paper. This step also means that the network is a self-establishing network, not to be confused with self-organizing networks, as the organization of the network only happens when a new node joins or leaves the network instead of the network having the ability to reorganize itself as expected from self-organizing networks. Self establishment is carried out by the coordination of all the nodes, as every node sends its own vote on clustering any other node. The decision to put the nodes into the best-fit cluster does not come from a singular node, but it is carried out according to the consensus of the entire network. During this process, however, the nodes may or may not find all the other nodes in the network. Which results in the network forming itself into what's called a Mesh Network, where the network participants are connected to the ones they have found. The number of nodes a given node can be connected to is also dynamic, as each node finds a different number of nodes in the network during this phase. There is also no hierarchical difference between the

nodes. All nodes are equal participants of the entire network with equal rights and privileges. Even if they are in different clusters with different connection qualities, all nodes together with their votes and mined blocks are treated.

3.2 Distributed Voted Hierarchical Clustering

SECBoN uses a distributed voted clustering, where each node appears with an approved cluster for themselves. However, other nodes also confirm the connection quality between alternative nodes and they also put them into a cluster in their own clustering process. These gathering results are treated as *votes*. The *votes* here are significant because the associates recognize the cluster instead of obsessing themselves onto the cluster they *prefer*. SECBON ensures the potential connection quality between the participants and to forbid any misbehaviour within the parties. *Voting* enables the peers to acquire information from the others. The peers are later placed into the clusters they were voted for by the majority of the network. This means that participant nodes could not deceive the network in order to obtain any sort of unwarranted advantage over the others. We can summarize the algorithm used for the clustering with the following Algorithm 1:

The clustering process of SECBoN is shown In Figure 1. *Node y* tries addresses in order to find as many network participants both inside and outside its cluster (*1. Node Discovery*. This is the first step to form the network. The nodes try to find other nodes in the network until the time limit is reached. Time limit only applies to a peer. A global timer is not kept, the peers keep a timer for themselves when they begin their own clustering process. They use this timer to measure how much time has passed for them, so that they can finish their discovery step when the time is up. After reaching the time limit, the nodes then begin the clustering step, which is shown in Figure 2, where each node tries to find the best cluster both for themselves and other peers by using a hierarchical clustering algorithm. After which they each

Algorithm 1 Voted Distributed Hierarchical Clustering

Input: address range
Output: distributed hierarchical clusters

```
1: while currentTime ≤ timeLimit do
2:     foundPeers ← findNewPeer()
3: end while
4: for i = 1; i < foundPeers.size() do
5:     for j = i; j < foundPeers.size() do
6:         eucDist ← calcEucDistBetweenPeers()
7:         peerConn[i][j] ← eucDist
8:     end for
9: end for
10: while clusterCount > trgtClusCountParameter do
11:     mergeUntilTargetNumberOfClustersAreLeft()
12: end while
13: for peer ∈ foundPeers do
14:     sendClusterResultToFoundPeers()
15: end for
16: for result ∈ receivedClusterResults do
17:     mostFreqClus ← findMostFreqClusForSelf()
18: end for
19: for peer ∈ allFoundPeers do
20:     broadcasttheclusteringdecisionstopeer()
21: end for
```

vote for the cluster they think is best suited for them. However, the other peers do not accept the voted clusters, therefore they also carry out the calculations, these calculations are based on the latency of the connections between the peers, in order to find the best suited clusters not just for themselves but for the other peers as well (*2. Find optimal cluster*). During this process they also vote for other peers so that any malicious peers aiming to manipulate the clustering procedure for their own gain can be stopped before they impact the clustering process, therefore all the nodes end up reaching a consensus on how the clusters are to be formed. After the *voting* process, the clusters are formed and the clustering information is propagated through the network so that all the peers are informed of which cluster they themselves and all the other peers are placed in. This is the third and last step of the clustering process of SECBoN and it is both necessary because if the nodes are not informed of

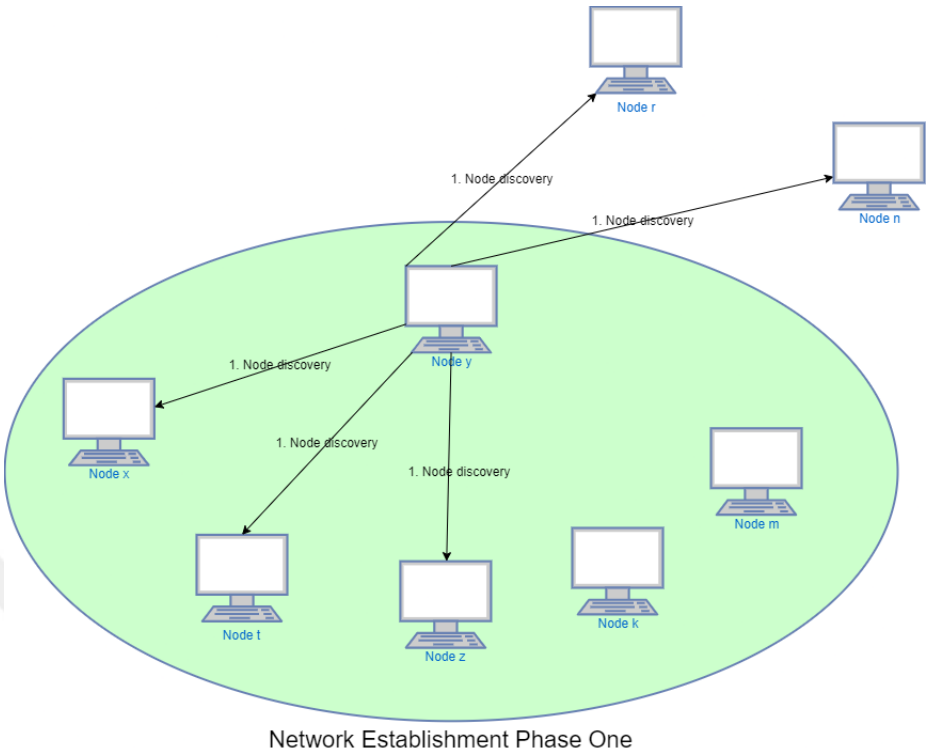

Figure 1: Phase 1 of the clustering procedure of SECBoN described on its network topology.

which cluster they were placed in by the consensus of the network then they cannot function inside the network and crucial in that regard because the *mining* process depends on the clusters to function as intended.

3.2.1 Mining Procedure

Through figures 3-5, the steps of the mining procedure are shown. Figure 3 shows the first two steps of the mining procedure. Step 1 is to mine blocks, in other words try to find a fitting hash value that is lower than of equal to that of the network difficulty for the block they are mining based on the contents of the said block, and add that block to the respective local chain of the cluster the nodes are a member of. When certain conditions are met, the local chain is then propagated firstly to the rest of the cluster by the node that has placed the last block into that local chain, which can be treated as Step 2. Afterwards the local chain is propagated to the nodes from other

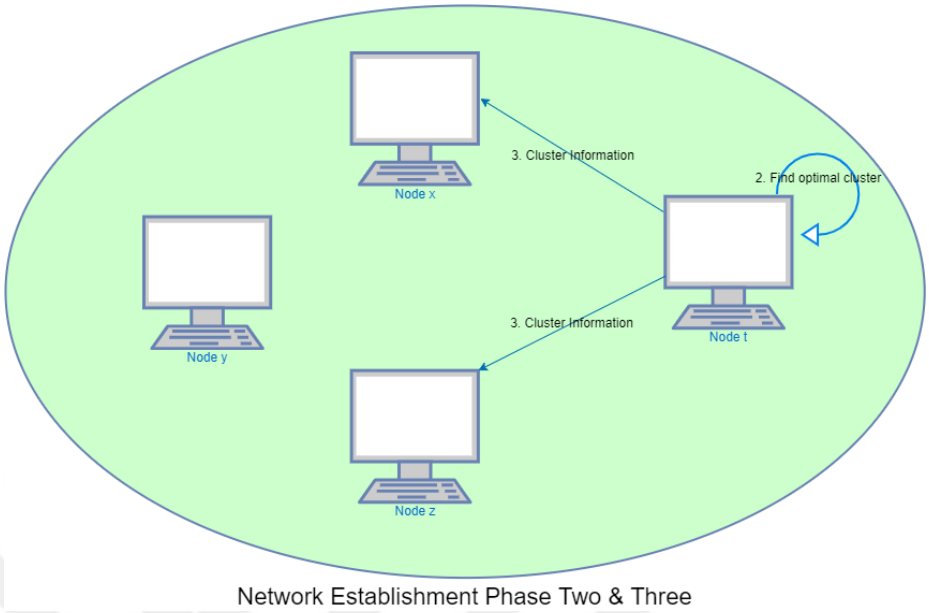

Figure 2: Phase 2 of the clustering procedure of SECBoN described on its network topology.

clusters as well. The nodes that receive the local chain also propagate it further down the network, which is Step 3. This ensures as much network coverage as possible for the propagation of the mined local chain and its verification thereof. Figure 4 shows the propagation of the local chain as it arrives into other clusters, the nodes pass along the local chain to the members of their own cluster first and then to all the other nodes they know within the network. Figure 5 shows the verification step, Step 4, which occurs when the time limit is reached for the propagation phase. During the verification procedure, the nodes verify the local chain they receive and then send the result back to the original owner of that local chain, in other words the node that had placed the last block into that local chain. When the time limit is reached, which is Step 5, the last node that receives the local chain verifies it and sends it back to the original miner as the last step of the mining procedure in SECBoN. When the original miner receives these verification results, it checks if the local chain has been verified by more than half of the network. If the majority of the network verifies the chain, the local chain is added to the global chain by the original miner of that local

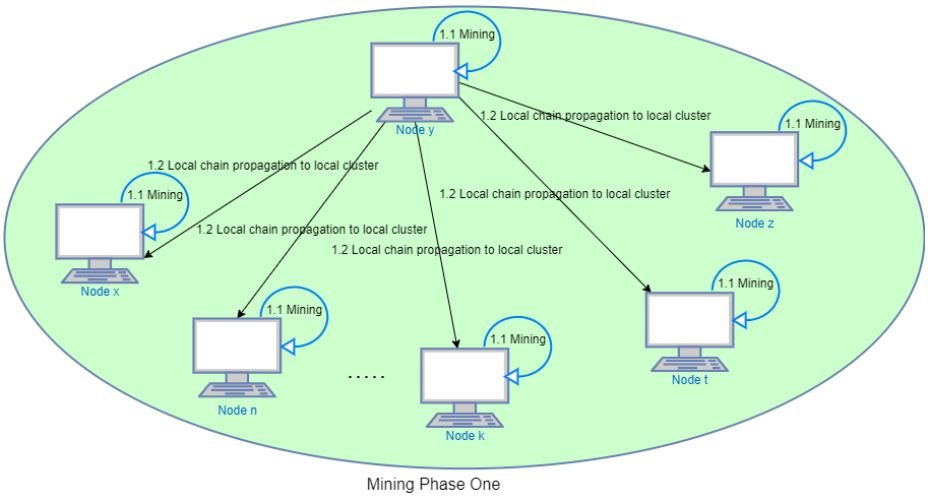

Figure 3: Phase 1 of the mining procedure of SECBoN described on its network topology.

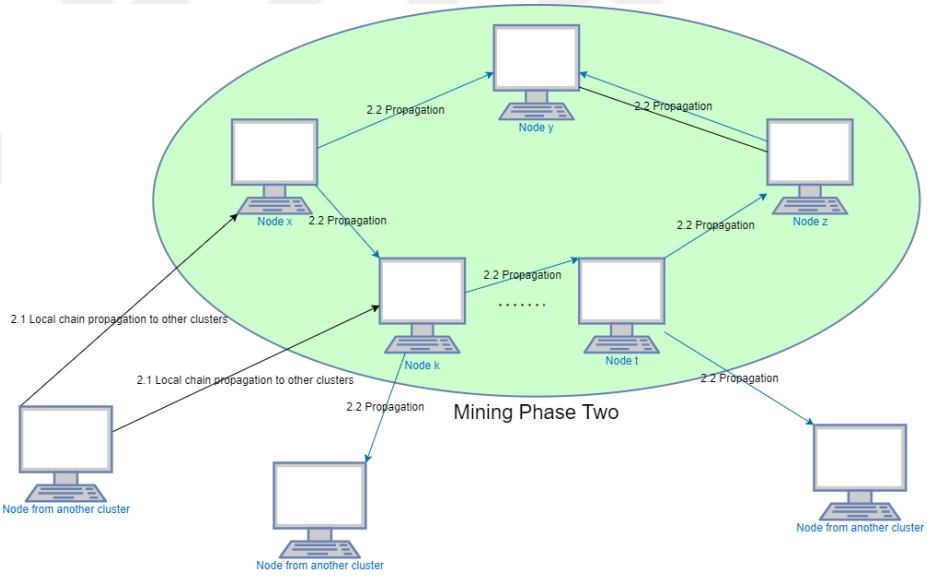

Figure 4: Phase 2 of the mining procedure of SECBoN described on its network topology.

chain. The original miner of a local chain is the node that could add the last block of the local chain.

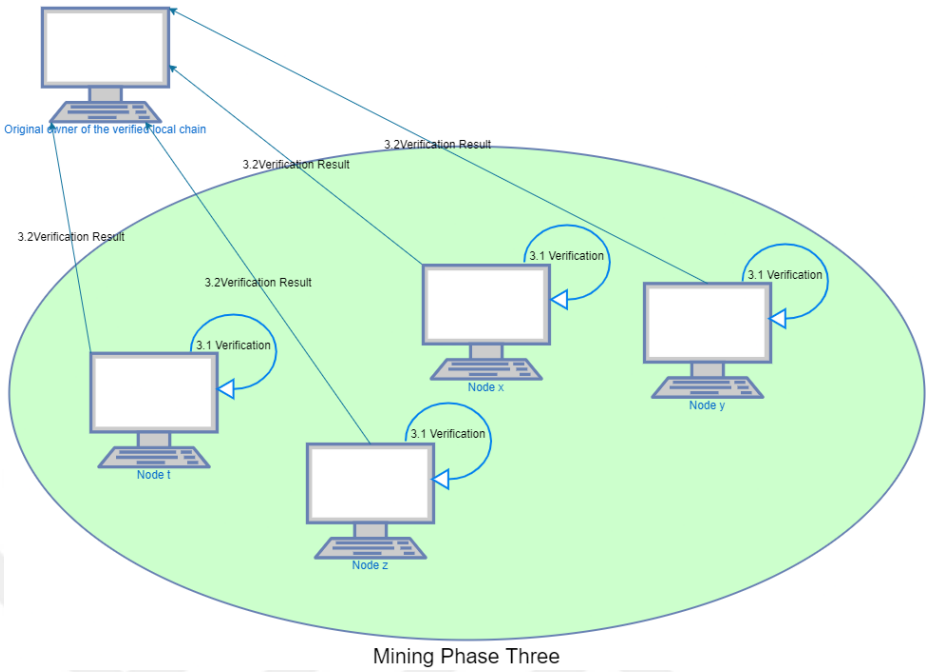

Figure 5: Phase 3 of the mining procedure of SECBoN described on its network topology.

3.3 Consensus Algorithm

The consensus algorithm used in SECBoN is a Proof of Work algorithm. SECBoN utilizes the basic principles of the Proof of Work algorithm of Bitcoin and applies the SHA256 hashing algorithm [2]. The nodes in the network are required to demonstrate the legitimacy of the work they present by trying to discover a hash value that is equal to or lower than the network difficulty. The contents of the blocks, such as the nonce, the details of the transactions in the transaction list and the previous block hash, all play a role in this endeavour where the nodes do mathematical calculations until they find a suitable hash value. The act of doing these mathematical calculations proves the work to be valid and the result of the calculations is acknowledged by the whole network.

3.4 Transactions

Transactions are abstractions of the data transfers from a participant of the network to another party. In [2], the transactions are used to describe and represent the electronic coins in the cryptocurrency network. The transactions are chained to the previous one by the participants' public keys, the hashed value of the previous transaction, and the signature of the owner of the previous transaction. Having these public keys and signatures in the transactions allows the network to see that a transaction was carried out between two valid participants.

During the development of SECBoN and the experimentation phase, the transactions have all consisted of randomized text, together with the hashed value of their contents. This was implemented in order to keep the development and experiments simple and efficient.

3.4.1 Local Chains and Global Chain

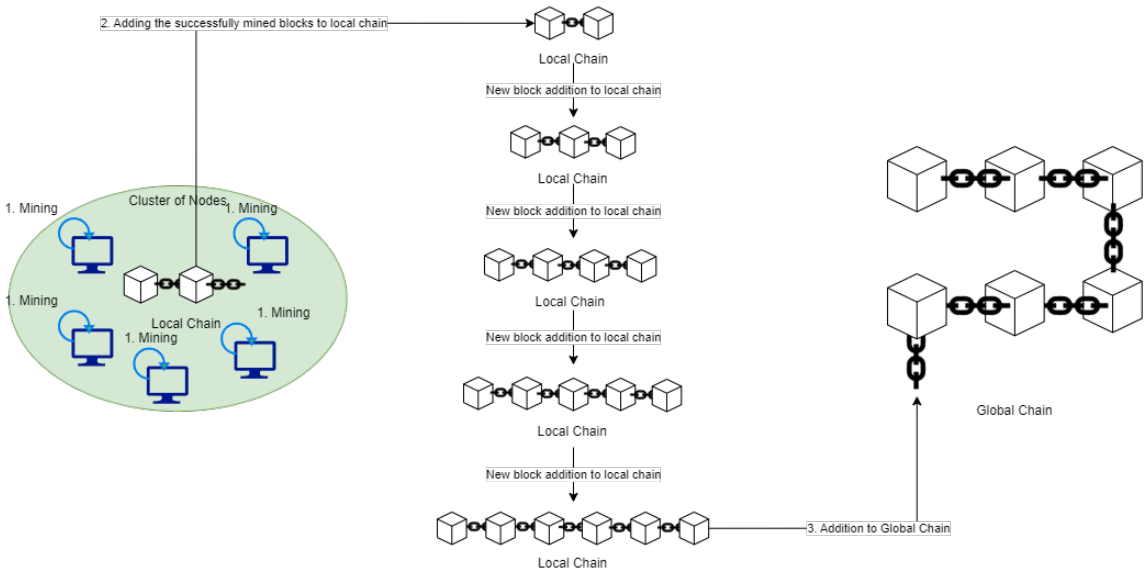

Figure 6: Local chain and Global chain usage visualization.

The clusters in the network mine their own *local chains* which are simply small sized blockchains, the length of which is parametrised. When these local chains reach the chain length during the mining phase, the node that finds the last block for a chain verifies it by itself first and then sends it first to the other nodes in its cluster and then to all other nodes it knows from other clusters as well in order to get the local chain verified. These local chains are sent with a timestamp, to mark the beginning of the verification process for that given local chain. Once the receiving nodes finish verifying the local chain they've received, they also propagate it over the network to get as much verification as possible from the network. Once a local chain is verified by a node, it is then marked and stored on a temporary list as verified by that node. This is needed so that the same node does not verify the same local chain again and again while it is getting propagated through the network. This verification process also has a set time limit, which also is another parametrised aspect of SECBoN, after which the nodes verify the relevant chain they receive and instead of propagating it further, they only send the verification result to the original sender of that local chain. Once the time is up and all the verification results are sent to the original sender of a local chain, they are then counted and if more than half of the network has verified that chain and found that it was a valid chain, the owner of that chain adds it to the actual global chain that the whole network shares.

However, the current network difficulty of SECBoN differs from Bitcoin's updated network difficulty. Bitcoin network changes its network difficulty at every 2016 blocks according to the average time to mine a block on the network. Whereas in SECBoN, the network difficulty has so far been static, meaning that there were no changes to the difficulty during the development or the experimentation processes. The main reason for this is because with a difficulty that is too low, the nodes mine the blocks too and the machine the nodes run on cannot handle all the communication the nodes try to do, therefore the actual performance of the program suffers. With a difficulty

that is too high, the mining process slows down, therefore reducing the transaction throughput of the network, since they are sharing the processing power of a single computer. A lower transaction throughput means a lower amount of blocks would be mined in the same amount of time compared to when the network uses a lower difficulty level. During the development and the experimentation processes, it was found that having six leading zeros on the block hashes provided the most desirable performance for testing and experimenting with the transaction rate of the network.

CHAPTER IV

PERFORMANCE EVALUATION AND RESULTS

In this section, simulation results are compared to measure the difference between SECBoN's architecture and Bitcoin in terms of the network transaction throughput performance. SECBoN and Bitcoin architectures are compared in the same platforms to have a more accurate performance comparison.

4.1 Network Parameters

Reconfiguration of particular features of the architecture is one of the essential parts of the structure. These specifications are essential for the optimization of the system performance. Defined parameters of the network that were applied during the development, experimentation and data collection processes are summarized in table:

Table 1: Fixed network parameters of SECBoN

Discovery time limit(ms)	Verification time limit(ms)	Propagation time limit(ms)	Propagation Node Count
300000	30000	30000	5

Parameters that remained unchanged during the development of SECBoN and the experimentation phase, where the network throughput data was collected are shown in Table I. The discovery time limit where the nodes try random addresses to find others is related to how well-connected the network is. A longer time limit for this step would cause the nodes finding even more nodes on the network, which would also mean a better connected network would be achieved. From the participant nodes'

perspective, better connected network means the propagation of local chains and other information would be easier. The nodes could cover a wider section of the network in less time, therefore increasing the network performance. However, more connections between the nodes would also mean an increased network traffic, creating even more congestion, therefore reducing the network performance. This duality means that parameter which is used during the network establishment phase, works both ways towards increasing or decreasing the network performance, so it must be set in such a way that its effects on the network increase the network performance instead of decreasing it. The chain difficulty paramater determines how many leading 0s the hash values of the mined blocks are supposed to have in order to be accepted into the global chain. During the development of SECBoN it was observed that setting this parameter too low results in too many blocks being mined and the network gets flooded with information, reducing its performance. This excess amount of mined blocks results in orphan blocks, even orphan local chains, as these chains are treated as a single unit when they are verified and propagated by and through the network. Just like Bitcoin, SECBoN also discards these orphan blocks, therefore having an excessive amount of orphan blocks also means wasted resources and time, therefore a reduced network performance is observed when the chain difficulty is set too low. Setting the value too high resulted in too few blocks being mined and accepted, therefore delaying the experiments, therefore an optimal of 6 leading zeros were used during development and experimentation. However, aside from slowing down the system, the chain difficulty does not have any other setbacks, in fact, it might even help reduce the network flooding even further, therefore allowing the nodes to communicate much more fluently, without having to wait for one another.

The second and third parameters set a limit for the propagated local chains to be verified and propagated even further through the network, only affects the percentage of the network that can verify the local chains they receive before the time runs

out. This time limitation also has both positive and negative effects on the network performance. One of the positive effects is that the limitation allows the nodes to stop propagating the local chains they receive, allowing the nodes themselves to reduce network congestion. Stopping the propagation of the local chains also means that the nodes with slow connections to the rest of the network would not have to worry about wasting their resources on either waiting on, receiving or sending information through their connections, therefore allowing the nodes to use their resources. However, on the downside, not all the nodes may get to verify and propagate the local chains, resulting in a reduced percentage of network coverage for the local chains meaning some chains may never get to be verified by more than fifty percent of the network, therefore getting discarded completely. This would mean that some of the local chains, meaning some of the mining effort on the network, would be lost forever, resulting in a reduced overall network performance. Meaning that these parameters need to be adjusted to maximize their positive effects and minimize their negative effects on the network to improve the throughput.

The fourth parameter decides the number of nodes each node propagates the local chains to after they verify one they have received before. This number limit reduces the network congestion by limiting the number of nodes each node sends information to during the verification and propagation phase. The nodes are selected at random from the list of known nodes to a given node, therefore the local chains are given a chance to reach as many nodes in the network as they can before the propagation time limit is reached. This way the propagation of the local chains is able to cover as much of the network as it can all the while limiting the flooding, and the congestion the flooding causes, that happens during this process. Increasing this number would also increase the network flooding, therefore increasing the network congestion as well and end up reducing the network performance. However setting this parameter too low would mean the local chains may not be propagated thoroughly through the

network and not being able to reach as many nodes as they would have been able to otherwise. In other words, this parameter affects the percentage of network coverage for the propagation of the local chains and the network congestion that comes with it, therefore, it also must be adjusted to maximize the network performance accordingly.

The parameters mentioned above are the parameters that were taken into consideration the most while developing SECBoN. These parameters were adjusted in order to find the most optimal values in order to increase the network performance.

4.2 Simulation Environment

The simulations of SECBoN were carried out on a single computer running Windows 10 Pro. The computer has a Ryzen 5 3600x CPU, so it has a great multi-threading performance. Multi-threading performance is important because all the simulated nodes were new threads that were run in parallel during the simulation of SECBoN. In order to allow the nodes to use as much processing power as possible during the testing phase, the number of background tasks that were running on the computer were kept to a minimum. This was done in order to achieve consistent results with no data fluctuations that would have occurred otherwise because of external reasons such as the background tasks.

There are three important metrics that were taken into account when assessing the performance of the networks mentioned in this thesis. The most obvious one is the transaction throughput of the network, the metric which depicts how many transactions the network is able to process through time. This metric shows the number of transactions processed in seconds against the size of the global chain. This makes it easy to assess which network is able to process the same number of transactions in a shorter amount of time. This metric is called throughput and it is measured in transactions per second (Tps). The remaining two metrics are for the latency of the connections between the participant nodes of the networks. One of

these metrics measures the latency values of the connections between the nodes that are used for the propagation of the mined blocks through the network. The other latency metric measures the latency for the nodes to be able to contact other nodes for the first time when they mine a local chain or a block so that they can send the mined data to the rest of the network in order to get the network to verify their mined data. This is quite an important metric since if the nodes were unable to contact one another when they mine a new local chain or a block, then the network would get stuck and no new verification or propagation would occur. These two remaining metrics are called Propagation Latency and Communication Latency, respectively.

4.3 Results

Numerous tests were conducted with SECBON, using the approach presented in this paper, to gather transaction throughput data with numerous parameters the network has such as the number of transactions included in each block mined and to evaluate their effects on the network's performance.

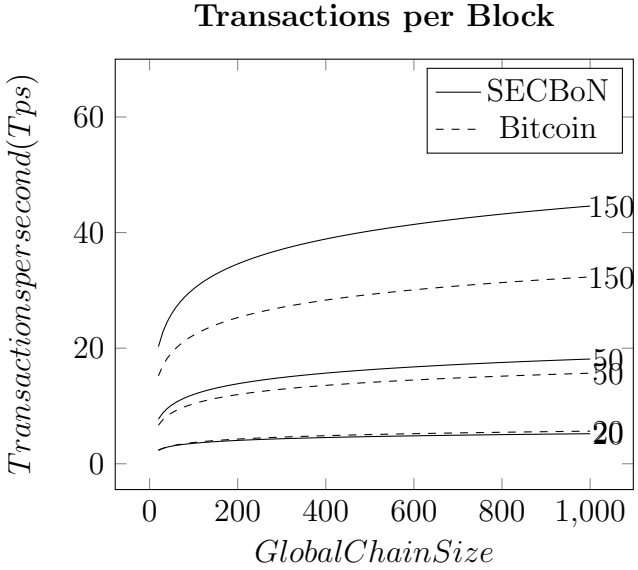

Figure 7: Network throughput difference between SECBoN and Bitcoin simulations when the number of transactions per block is set to 20, 50 and 150, respectively.

Figure 7 shows the comparison of the measured network throughput tendencies

of SECBoN and Bitcoin simulations when the number of transactions included in each block is adjusted gradually, in transactions per second. The X-axis of the graph depicts the number of blocks which have been verified and added to the global chain of the network and the Y-axis represents the transaction throughput of the network measured in transactions per second (Tps) The dashed lines represent the Bitcoin simulation and the whole lines represent SECBoN's performances, respectively. The data show a decent improvement in the Tps metric of the network as more transactions are included in each block mined. The higher the throughput at any given global chain length achieved, the better the performance of the network. Meaning that SECBoN tends to outperform Bitcoin in terms of the network throughput as it is able to handle more transactions in the same amount of time than Bitcoin. This outcome is the result of the way SECBoN makes use of clustering for its nodes and how it handles the inter-node communications both within the clusters and between the clusters of the network. By making clever use of the network resources of the nodes, SECBoN is able to utilize the potential of the nodes effectively in such a way that the nodes do not become overwhelmed with excessive communication requests from the other participants, therefore the effect of the disruption of the data flow between the nodes is kept to a minimum. Since the nodes make effective use of their resources, do not get overwhelmed and are able to effectively contribute to the collective effort of the network in order to achieve better throughput, SECBoN tends to outperform Bitcoin[2]. Having lower latency values in the propagation phase for the mined local chains also means that the network coverage will also be higher as the nodes will be able to reach more participants in the same amount of time, which improves the security of the system as well as its transaction processing performance. The data shown in Figure 7 prove that SECBoN successfully makes use of clustering and reconfiguration of certain network elements to increase the overall performance in terms of the transactions processed by the network.

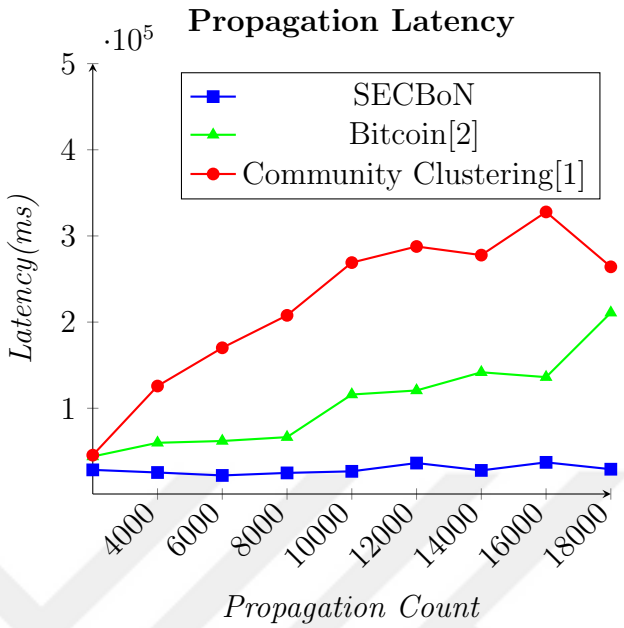

Figure 8: Latency difference of the propagation of received chains by the nodes on having clusters against not having clusters (Propagation Latency)

Figure 8 shows the average latency values of SECBoN, Bitcoin[2] and Community Clustering[1] during the propagation phase where the mined local chains are being propagated through the network in order for the participant nodes to verify them. In this case, lower latency values indicate a better network performance. Which means SECBoN, having the lowest average latency values at all points shows that it's the best out of all the simulations at handling the network flooding and node-to-node communications for the propagation of the mined local chains. Which also means, SECBoN achieves the best network coverage in a given amount of time since the nodes in the other simulations, Community Clustering[1] and Bitcoin respectively, clearly fall behind and are unable to communicate as effectively. This is the most clear with the Community Clustering[1] where the average latency is the highest, one of the factors for this being the case is the fact that some of the nodes in such a clustering scheme are selected to be members of at most 2 clusters. Which also means for a given local chain to arrive in another cluster, it first will have to go through this node that is a shared member between two different clusters. This approach alone

puts a lot more communications stress on any given shared node in the network, flooding it with the processing amount of two whole clusters of nodes. This in turn causes a much higher overall latency inside the network as the nodes will have to wait for the shared member to process all of its incoming and outgoing communications. This network congestion is even beaten by the Bitcoin simulation where the nodes try to communicate freely, instead of having to go through a shared member as can be seen in the graph even the Bitcoin simulation has lower average latency values than Community Clustering[1]. In SECBoN, however, this is not the case and the participant nodes are able to make use of their resources in a much more effective way since the nodes do not have to go through any given shared member between clusters when propagating mined chains, instead, they are able to freely reach out to any node they have found previously. Therefore, due to its better handling of network congestion, SECBoN is able to provide a lower, hence better, overall network latency which in turn allows SECBoN to process more transactions per second compared to the other simulations.

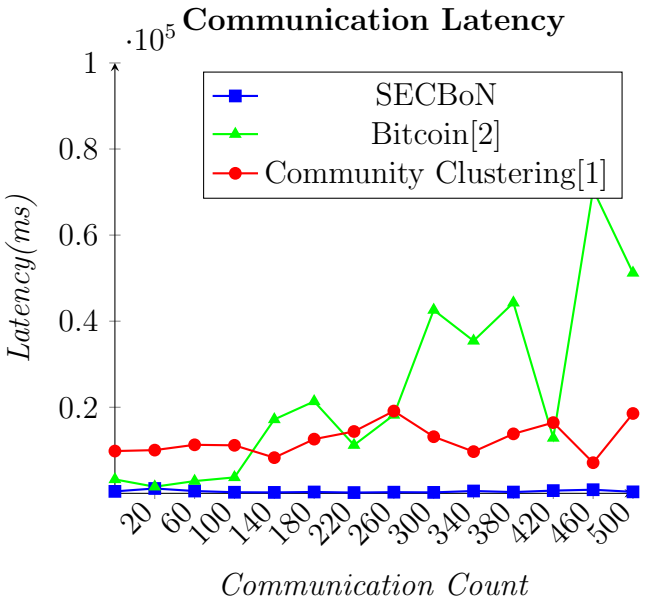

Figure 9: Latency difference of the step where the nodes begin the communication process in order to send their mined local chains to the rest of the network on having clusters and not having clusters (Communication Latency)

Figure 9 shows the latency values of when the nodes try to establish a connection to another node so that the propagation and verification process of the local chain they mined may begin. This step is crucial as it is the beginning of the verification process for all the blocks mined by the participants of the network. As higher latency values will cause delays in the verification process, having lower latency values means that the network performs better compared to the others. With that in mind, looking at the values shown in the graph, which are measured during the simulations of SECBoN, Bitcoin[2] and Community Clustering[1] respectively, it is easy to see that SECBoN, having the lowest latency values out of all the simulations proves that it provides a better communication environment for its participant nodes so that the nodes have a greater ability to communicate with the rest of their network in order to get the local chains they mined verified by the other members of the network. The worst latency values are shown by the Bitcoin simulation as expected since the nodes act in a chaotic manner and may cause unnecessary flooding in their respective network, this flooding is further reinforced by the fact that the nodes in the Bitcoin network try to send every single block they mine instead of local chains, therefore the number of connections being made is much higher, therefore the nodes become flooded with communication requests in a shorter amount of time. This means that the way the communications between the nodes is handled affects the performance of the system drastically as higher latency values means that the nodes are unable to communicate as effectively compared to the nodes in the SECBoN network.

When the data representing SECBoN are compared to the data acquired from Bitcoin and Community Clustering[1] simulations, SECBoN outperforms both of the other proposed architectures both in terms of the network transaction throughput and the average network latency. SECBoN tends to provide better transaction throughput, tends to perform under lower latency values both during the propagation phase and the first communication needed to begin the verification phase of the mined local

chains. The data show that SECBoN achieves its goal of allowing a better usage of network resources by providing lower average latency values overall through its communication steps and by providing a higher Tps metric compared to Bitcoin and Community Clustering[1].

CHAPTER V

CONCLUSIONS

In this thesis, we introduced a self-establishing clustered blockchain network architecture which provides a hierarchical structure. The proposed architecture satisfies the needs of a number of issues such as scalability, performance, data privacy and security and auditability. This is achieved by the mechanisms proposed in this thesis that allow the network participants to use their resources more efficiently and effectively. The efficient and effective use of the network resources is achieved by the use of local chains that are used by their respective clusters in the network for the cluster members to mine on. Doing so reduces the number of network connections and provides a form of traffic handling which reduces the communications load on the participant nodes. Reduced communications load on the participant nodes results in improved performance as the nodes will be able to use more of their processing power on mining and verification instead. The performance of the architecture proposed is evaluated based on three metrics. The most prominent metric is the transaction throughput which depicts the number of transactions processed per second. The second metric, called propagation latency which assesses the effects of the flooding that occurs during the propagation phase of the mined blocks. This metric depicts how well and quickly the participants are able to communicate with one another in order to achieve as much coverage as possible for the blocks to be verified by the network. Lower latency values depict a better communications environment for the participants. The third metric, called communication latency, assesses how easily the mined blocks are able to begin their verification process throughout the network. The longer the participants take in order to be able to send their mined blocks to others in the network for the

first time for any given block, the worse the network performance will be in terms of transaction throughput. In other words, beginning the propagation and verification phase for the mined blocks as fast as possible is crucial for the performance of the network. If the flooding of the network is not handled properly, it will cause delays in communications and poor network performance as a result. SECBoN handles the network flooding by the use of local chains that are used to send the mined blocks in groups rather than a single block at a time for propagation and verification. This method minimizes the flooding effect on the communications between the participants. The proposed work is compared to Bitcoin[2] and to Community Clustering[1] based on the latency metrics mentioned and results suggest that SECBoN surpasses Bitcoin[2] in terms of transaction throughput and surpasses Bitcoin[2] and Community Clustering[1] both in terms of average latency of the network by providing much lower overall network latency values. SECBoN achieves its goals and provides improvements upon previously proposed clustered blockchain networks by providing a network environment in which the participants are able to use their resources more effectively. As a future work, SECBoN can be implemented on actual testing environments with participant computers that are physically located in different places so that a real use-case environment can be achieved for more accurate testing.

Bibliography

[1] S. Li *et al.*, "Blockchain dividing based on node community clustering in intelligent manufacturing cps," in *IEEE International Conference on Blockchain (Blockchain)*, pp. 124–131, 2019.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." https://bitcoin.org/bitcoin.pdf (accessed: 26.08.2021).

[3] W. Chen *et al.*, "A survey of blockchain applications in different domains," in *Proceedings of the International Conference on Blockchain Technology and Application*, ICBTA, (New York, NY, USA), p. 17–21, Association for Computing Machinery, 2018.

[4] Q. E. Abbas and J. Sung-Bong, "A survey of blockchain and its applications," in *International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 001–003, 2019.

[5] M. Krichen *et al.*, "Blockchain for modern applications: A survey.," *Sensors (14248220)*, vol. 22, no. 14, p. N.PAG, 2022.

[6] K. Zahoor *et al.*, "Blockchain applications for covid-19-a survey.," *Suranaree Journal of Science Technology*, vol. 29, no. 5, pp. 1 – 13, 2022.

[7] Y. Abuidris *et al.*, "A survey of blockchain based on e-voting systems," in *Proceedings of the 2nd International Conference on Blockchain Technology and Applications*, ICBTA, (New York, NY, USA), p. 99–104, Association for Computing Machinery, 2020.

[8] C. Wang *et al.*, "A survey: applications of blockchain in the internet of vehicles.," *EURASIP Journal on Wireless Communications Networking*, vol. 2021, no. 1, pp. 1 – 16, 2021.

[9] Z. Zheng *et al.*, "An overview of blockchain technology: Architecture, consensus, and future trends," in *IEEE International Congress on Big Data (BigData Congress)*, pp. 557–564, 2017.

[10] L. Lao *et al.*, "A survey of iot applications in blockchain systems: Architecture, consensus, and traffic modeling," *ACM Comput. Surv.*, vol. 53, feb 2020.

[11] A. A. Monrat *et al.*, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117134–117151, 2019.

[12] X. Wang *et al.*, "Survey on blockchain for internet of things," *Computer Communications*, vol. 136, pp. 10–29, 2019.

[13] W. Gao *et al.*, "A survey of blockchain: Techniques, applications, and challenges," in *27th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–11, 2018.

[14] D. Berdik *et al.*, "A survey on blockchain for information systems management and security," *Information Processing Management*, vol. 58, no. 1, p. 102397, 2021.

[15] M. Fadhil *et al.*, "A bitcoin model for evaluation of clustering to improve propagation delay in bitcoin network," in *IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, pp. 468–475, 2016.

[16] H. Jin *et al.*, "Cross-cluster federated learning and blockchain for internet of medical things," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[17] A. Biryukov and S. Tikhomirov, "Transaction clustering using network traffic analysis for bitcoin and derived blockchains," in *IEEE INFOCOM - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 204–209, 2019.

[18] K. Saadat *et al.*, "Reconfigurable blockchains for dynamic cluster-based applications," in *IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pp. 925–931, 2020.

[19] M. J. Islam *et al.*, "Blockchain-sdn based energy-aware and distributed secure architecture for iots in smart cities," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[20] G. P. o. Joshi, "Toward blockchain-enabled privacy-preserving data transmission in cluster-based vehicular networks," *Electronics*, vol. 9, no. 9, 2020.

[21] H. o. Honar Pajooh, "Multi-layer blockchain-based security architecture for internet of things," *Sensors*, vol. 21, no. 3, 2021.

[22] J. Mittendorfer and K. A. Hummel, "Mitigating messaging and processing load in iot environments managed by blockchain," in *Proceedings of the 17th International Conference on Advances in Mobile Computing amp; Multimedia*, MoMM, (New York, NY, USA), p. 231–235, Association for Computing Machinery, 2019.

[23] H. Pajooh *et al.*, "Multi-layer blockchain-based security architecture for internet of things," *Sensors*, vol. 21, no. 3, 2021.

[24] M. J. Amiri *et al.*, "Sharper: Sharding permissioned blockchains over network clusters," in *Proceedings of the International Conference on Management of*

Data, SIGMOD '21, (New York, NY, USA), p. 76–88, Association for Computing Machinery, 2021.

[25] M. Fadhil *et al.*, "Locality based approach to improve propagation delay on the bitcoin peer-to-peer network," in *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 556–559, 2017.

VITA

Orkun Doğan is a graduate student in the Computer Science Department of Özyeğin University in Turkey. He received B.Sc. degree from Özyeğin University in 2019. His current research interests include security and scalability for P2P networks.